databricks

Databricks 試験ガイド

Databricks Certified Data Engineer Associate

試験ガイドについてフィードバックを送る

この試験ガイドの目的

この試験ガイドでは、試験の準備に役立てていただくために試験の概要と試験の対象範囲について説明します。試験になんらかの変更がある場合には（そして、それらの変更が試験に反映される際には）、試験の準備を行えるように本書の内容は随時更新されます。本バージョンは、**2024** 年 **1** 月 **1** 日現在の実施試験に対応しています。試験を受ける **2** 週間前に、ご自身の試験ガイドが最新版であることを再度ご確認ください。

対象者についての説明

Databricks Certified Data Engineer Associate 認定試験では、Databricks レイクハウスプラットフォームを使用して入門レベルのデータエンジニアリングタスクを完了する個人の能力を評価します。これには、レイクハウスプラットフォームとそのワークスペース、アーキテクチャ、機能についての理解が含まれます。また、バッチ処理と増分処理、両方のパラダイムで Apache Spark SQL とPython を使用してマルチホップアーキテクチャ ETL タスクを実行する能力も評価します。さらに、受験者はエンティティの権限を維持しながら基本的な ETL パイプラインと Databricks SQL クエリーおよびダッシュボードを本番運用に導入する能力を評価されます。この認定試験に合格した人は、Databricks とその関連ツールを使って基本的なデータエンジニアリングタスクを完了できると期待されます。

試験の概要

- 問題数: 45 問の多肢選択問題
- 制限時間: 90 分
- 受験料: 200 米ドル。および、現地の法律によって適用される税金が加算されます
- 実施方法: オンライン（監督付き）
- 試験での持ち込み: 一切利用できません。
- 前提条件: なし。Databricks に関するコースの受講と 6 か月の実務経験が強く推奨されます
- 有効期間: 2 年間
- 採点対象外の内容: 今後使用する統計情報を収集するために、試験には採点対象外の項目が含まれている場合があります。これらの項目は、フォームでは特定されず、得点には影響しません。この内容については、追加の時間が考慮されています。

推奨されるトレーニング

- インストラクター主導: Data Engineering with Databricks
- セルフペース: Data Engineering with Databricks（Databricks Academy で提供）

試験の概要

セクション **1: Databricks** レイクハウスプラットフォーム

- データレイクハウスとデータウェアハウスの関係を説明する。
- データレイクとの比較で、データレイクハウスにおけるデータ品質の改善点を特定する。
- シルバーテーブルとゴールドテーブルを比較対照し、どのワークロードでソースとしてブロンズテーブルを使用し、どのワークロードでソースとしてゴールドテーブルを使用するのかを判断する。
- データプレーンとコントロールプレーンに配置される要素や、顧客のクラウドアカウント内に存在する要素など、Databricks プラットフォームアーキテクチャに含まれる要素を特定する
- All-Purpose クラスターとジョブクラスターの違いを理解する。
- Databricks Runtime を使用したクラスターソフトウェアのバージョン管理方法を特定する。
- クラスターをフィルタ処理して、ユーザーがアクセス可能なものを表示する方法を特定する。
- クラスターが終了する仕組みと、クラスターの終了が及ぼす影響を説明する。
- クラスターの再起動が役に立つシナリオを特定する。
- 同じノートブック内で複数の言語を使用する方法について説明する。
- あるノートブックを別のノートブック内から実行する方法を特定する。
- ノートブックを他人と共有する方法を特定する。
- Databricks Repos が Databricks 内で CI/CD ワークフローを実現する方法について説明する。
- Databricks Repos 経由で利用可能な Git の操作を特定する。
- Repos との比較で Databricks ノートブックのバージョン管理機能にどのような制限があるかを特定する。

セクション **2: Apache Spark** での **ELT**

- 単一のファイルからのデータ抽出と、複数のファイルを含むディレクトリからのデータ抽出を行う
- FROM キーワードの後ろにデータタイプとして含まれている接頭辞を特定する。
- ビュー、一時ビュー、CTE をファイルの参照として作成する
- 外部ソースからのテーブルが Delta Lake テーブルでないことを特定する。
- JDBC 接続と外部 CSV ファイルからテーブルを作成する
- count_if 関数の使用方法と、x が null のカウントの使用方法を特定する
- count(row) で NULL の値をスキップする方法を特定する。
- 既存の Delta Lake テーブルから行の重複を排除する。
- 既存のテーブルから重複する行を削除して新しいテーブルを作成する。
- 特定の列に基づいて行の重複を排除する。
- すべての行に対してプライマリキーが一意であることを確認する。
- フィールドが別のフィールド内の一意の値に 1 つだけ関連付けられていることを確認する。
- 値が特定のフィールドにないことを確認する。
- 列をタイムスタンプにキャストする。
- タイムスタンプからカレンダーのデータを抽出する。

- 既存の文字列列から特定のパターンを抽出する。
- ドット構文を利用して、ネストされたデータフィールドを抽出する。
- array 関数を使用するメリットを特定する。
- JSON 文字列を解析して構造体にする。
- 結合クエリーに基づいて返される結果を特定する。
- explode 関数と flatten 関数を使用するシナリオを特定する
- ワイド形式からロング形式にデータを変換する手段として PIVOT 句を特定する。
- SQL UDF を定義する。
- 関数の場所を特定する。
- SQL UDF を共有するためのセキュリティモデルについて説明する。
- SQL コードに CASE/WHEN を使用する。
- カスタム制御フローに CASE/WHEN を活用する。

セクション **3:** 増分データ処理

- Delta Lake が ACID トランザクションを提供する場所を特定する
- ACID トランザクションのメリットを特定する。
- トランザクションが ACID に準拠しているかどうかを特定する。
- データとメタデータを比較対照する。
- マネージドテーブルと外部テーブルを比較対照する。
- 外部テーブルを使用するシナリオを特定する。
- マネージドテーブルを作成する。
- テーブルの場所を特定する。
- Delta Lake ファイルのディレクトリ構造を調べる。
- 前のバージョンのテーブルを記述した人物を特定する。
- テーブルトランザクションの履歴を確認する。
- テーブルを前のバージョンにロールバックする。
- テーブルを前のバージョンにロールバックできることを特定する。
- 特定のバージョンのテーブルをクエリーする。
- Z-Ordering が Delta Lake テーブルで有益な理由を特定する。
- VACUUM でどのようにして削除がコミットされるのかを特定する。
- OPTIMIZE によって圧縮されるファイルの種類を特定する。
- CTAS をソリューションとして特定する。
- 生成された列を作成する。
- テーブルのコメントを追加する。
- CREATE OR REPLACE TABLE と INSERT OVERWRITE を使用する
- CREATE OR REPLACE TABLE と INSERT OVERWRITE を比較対照する
- MERGE を使用すべきシナリオを特定する。
- 保存時にデータの重複を排除するコマンドとして MERGE を特定する。
- MERGE コマンドのメリットを説明する。
- COPY INTO ステートメントではターゲットテーブルのデータの重複が排除されない理由を特定する。
- COPY INTO を使用すべきシナリオを特定する。

- COPY INTO を使用してデータを挿入する。
- 新しい DLT パイプラインを作成するために必要なコンポーネントを特定する。
- パイプライン作成におけるターゲットとノートブックライブラリの目的を特定する。
- コストとレイテンシーの観点から、トリガー式のパイプラインと継続的なパイプラインを比較対照する
- Auto Loader を利用しているソースの場所を特定する。
- Auto Loader が有益なシナリオを特定する。
- Auto Loader では JSON ソースから推論されたデータがすべて STRING になる理由を特定する
- 制約違反のデフォルト動作を特定する
- 制約違反に対する ON VIOLATION DROP ROW と ON VIOLATION FAIL UPDATE の影響を特定する
- チェンジデータキャプチャと、APPLY CHANGES INTO の動作について説明する
- イベントログをクエリーして、メトリクスの取得、監査ロギング、リネージの検査を行う。
- DLT 構文のトラブルシューティング: エラーを発生させた DLT パイプライン内のノートブックを特定する。CREATE ステートメントに LIVE が必要であることを特定する。FROM 句に STREAM が必要であることを特定する。

セクション **4:** 本番運用パイプライン

- ジョブで複数のタスクを使用するメリットを特定する。
- ジョブで先行タスクを設定する。
- 先行タスクを設定すべきシナリオを特定する。
- タスクの実行履歴を確認する。
- スケジューリングの機会として CRON を特定する。
- 失敗したタスクをデバッグする。
- 失敗時の再試行ポリシーを設定する。
- タスクが失敗したときのアラートを作成する。
- E メールで送信できるアラートを特定する。

セクション **5:** データガバナンス

- データガバナンスの 4 領域のいずれかを特定する。
- メタストアとカタログを比較対照する。
- Unity Catalog でセキュリティの設定が可能な内容を特定する。
- サービスプリンシパルを特定する。
- Unity Catalog と互換性がある、クラスターのセキュリティモードを特定する。
- UC が有効な All-Purpose クラスターを作成する。
- DBSQL ウェアハウスを作成する。
- 3 層の名前空間に対してクエリーを実行する方法を特定する。
- データオブジェクトのアクセスコントロールを実装する
- メタストアとワークスペースを同じ場所に配置することがベストプラクティスであると特定する。
- サービスプリンシパルを接続に使用することがベストプラクティスであると特定する。
- カタログ全体で事業部門を分けることがベストプラクティスであると特定する。

サンプル問題

これらの問題は旧バージョンの試験から削除されたもので、試験ガイドに記載されている目的を示し、各目的に対応するサンプル問題を提示することを意図としています。試験ガイドには、試験の出題対象になる可能性がある目的の一覧が記載されています。認定試験の準備を行う際には、この試験ガイドの「試験の概要」を確認することをお勧めします。

問題 **1**
目的: 従来のデータウェアハウスとの比較で、データレイクハウスのメリットを説明する。

従来のデータウェアハウスにはない、データレイクハウスのメリットは何ですか。

A. データレイクハウスは、データ管理のリレーショナルシステムを提供する。
B. データレイクハウスは、バージョン管理のためにスナップショットを収集する。
C. データレイクハウスは、ストレージとコンピュートを統合して完全な管理を実現する。
D. データレイクハウスは、独自のストレージフォーマットをデータに利用する。
E. データレイクハウスは、バッチ分析とストリーミング分析の両方に対応している。

問題 **2**
目的: クエリーの最適化手法を特定する。

データエンジニアリングチームは、Delta テーブルに対してクエリーを実行して、同じ条件を満たす行をすべて抽出する必要があります。しかし、チームはクエリーの実行が遅いことに気付きました。データファイルのサイズは既に調整してあります。調査の結果、チームは、条件を満たす行が各データファイルの全体にまばらに存在すると結論付けました。

クエリーの速度を上げられるのは、どの最適化手法ですか。

A. データスキップ
B. Z-Ordering
C. ビンパッキング
D. Parquet ファイルとして記述
E. ファイルサイズの調整

問題 **3**
目的: シルバーテーブルをソースとして利用するデータワークロードを特定する。

シルバーテーブルをソースとして利用するデータワークロードはどれですか。

A. タイムスタンプを解析して人間が読み取れる形式にすることで、データをエンリッチ化するジョブ
B. ダッシュボードに既にフィードされている集計済みデータに対してクエリーを実行するジョブ
C. ストリーミングソースからレイクハウスに生データを取り込むジョブ
D. クリーニングしたデータを集計して、標準的なサマリー統計を作成するジョブ
E. 異常な形式のレコードを削除してデータをクリーニングするジョブ

問題 **4**
目的: 更新スケジュールの構成方法を説明する

エンジニアリングマネージャーは、顧客から報告があったバグを修正するチームの進捗状況を監視するために、Databricks SQL クエリーを使用しています。マネージャーはクエリーの結果を毎日チェックしていますが、その日ごとにクエリーを手動で再実行して、結果が返ってくるまで待っています。

クエリーの結果が毎日更新されるようにするには、クエリーをどのようにスケジュールすべきですか。

A. Databricks SQL のクエリーのページで 12 時間ごとに更新する。
B. Databricks SQL のクエリーのページで 1 日ごとに更新する。
C. ジョブの UI で 12 時間ごとに実行する。
D. Databricks SQL の SQL ウェアハウスのページで 1 日ごとに更新する。
E. Databricks SQL の SQL ウェアハウスのページで 12 時間ごとに更新する。

問題 **5**
目的: 適切な権限を付与するためのコマンドを特定する

ある会社で新しいデータエンジニアが働き始めました。会社の Databricks ワークスペースに、つい先日、このデータエンジニアが new.engineer@company.com として追加されました。データエンジニアは retail データベースの sales テーブルに対してクエリーを実行できる必要があり、retail データベースに対する USAGE 権限が既に付与されています。

新しいデータエンジニアに適切な権限を付与するには、どのコマンドを使用するべきですか。

A. GRANT USAGE ON TABLE sales TO new.engineer@company.com;
B. GRANT CREATE ON TABLE sales TO new.engineer@company.com;
C. GRANT SELECT ON TABLE sales TO new.engineer@company.com;
D. GRANT USAGE ON TABLE new.engineer@company.com TO sales;
E. GRANT SELECT ON TABLE new.engineer@company.com TO sales;

答え

問題 1: E
問題 2: B
問題 3: D
問題 4: B
問題 5: C